

Aplicações tecnológicas

Desenvolvimento de um *Software* aberto para cálculo de zeros de funções

Development of an open source software for calculating zeros of functions

David Lucas Pereira ¹ , Emerson Fouchy Santos ¹ 

¹ Universidade Federal de Santa Maria, Cachoeira do Sul, RS, Brasil

RESUMO

Este trabalho visa apresentar o software educacional Zero's Program, desenvolvido utilizando a linguagem de programação Python (backend), em conjunto com a biblioteca GTK, que se comunica com o frontend. O programa consiste em uma calculadora de zeros de funções, que por meio de uma interface gráfica, permite o usuário inserir os dados de entrada (função e local de análise), e, após, escolher os métodos de análise desejados. Se o usuário não conhece o comportamento da função, é possível gerar o seu gráfico, tornando-se mais simples a escolha de um intervalo de análise. Os métodos numéricos suportados no programa são o método da Bisseção, Newton-Rhapson, Posição Falsa e Birge-Vieta. Também é possível gerar uma curva de convergência para cada método, gerando-se uma maior compreensão do usuário aos métodos. Além disso, o software é de código aberto (open source), sendo possível, baixar e estudar como o mesmo funciona e sugerir modificações.

Palavras chave: Zero's Program; Python; GTK; Métodos numéricos; Código aberto

ABSTRACT

This work aims to present the educational software Zero's Program, developed using the Python programming language (backend), together with the GTK library, which communicates with the frontend. The program consists of a function zero calculator, which, through a graphical interface, allows the user to enter input data (function and analysis location), and then choose the desired analysis methods. If the user does not know the behavior of the function, it is possible to generalize its graph, making it simpler to choose an analysis interval. The numerical methods supported in the program are the Bisection method, Newton-Rhapson, False Position and Birge-Vieta. It is also possible to generate a convergence curve for each method, generating a greater understanding of the methods by the user. In addition, the software is open source, making it possible to download and study how it works and suggest modifications.

Keywords: Zero's Program; Python; GTK; Numerical methods; Open source



1 INTRODUÇÃO

Zeros de função, também conhecidos como raízes, são os pontos em que uma função cruza o eixo x , sendo $f(x)$ igual a zero. Encontrar os zeros de uma função é um desafio em áreas da matemática, da engenharia e das ciências, pois muitos problemas práticos requerem encontrar soluções para equações não lineares [1]. Para encontrar os zeros de uma função, podem ser usadas técnicas, sendo elas chamadas de métodos numéricos. O presente trabalho, utilizou métodos como o método da Bisseção, Newton-Raphson (NR), Posição Falsa (PF) e Birge-Vieta (BV).

O método da bisseção é um método de busca binária, em que, divide-se repetidamente o intervalo que contém a raiz em dois subintervalos de igual comprimento (média simples) e, em seguida, determina em qual subintervalo a raiz está localizada. Por outro lado, o método da posição falsa usa os mesmos conceitos de funcionamento da bisseção, mas, utiliza-se uma média ponderada. Já o método de Newton-Raphson, é um método iterativo que usa uma aproximação inicial para a raiz e juntamente com a derivada da função para melhorar/acelerar a aproximação em cada iteração. Em consonância, o método de Birge-Vieta é uma variação do método de Newton-Raphson, que usa um polinômio auxiliar para modificar a velocidade de convergência. Este método é somente válido para polinômios [3].

A implementação destes métodos pode ser feita por meio de *softwares* matemáticos como *Octave*, *Matlab* e *Maple*. Porém, também é possível utilizar linguagens de programação de alto nível para aplicar estes métodos, tais como *Python*, *C#*, *Javascript*, entre outros. *Python* é uma linguagem de programação, de fácil codificação, e que, possui uma grande variedade de bibliotecas e ferramentas disponíveis, como as bibliotecas *NumPy* e *Matplotlib*. A linguagem *Python*, também permite, através da biblioteca *GTK (Gimp Toolkit)*, que sejam criadas interfaces gráficas (GUI's) completas, geralmente utilizadas em programas em geral [2].

A biblioteca GTK consiste em um conjunto de ferramentas de código aberto,



que servem principalmente para criar interfaces gráficas. Sendo dividido em 3 camadas de bibliotecas para facilitar a portabilidade para outras plataformas, GLib, GDK e GTK, sendo que a terceira camada é independente de plataforma [4]. Os nomes das funções e dos tipos definidos nessas bibliotecas são descritivos e sempre possuem um prefixo para identificar a biblioteca [C]. O Glade é um designer de interface gráfica, que pode ser usado com a biblioteca GTK. Ele permite que o usuário crie interfaces gráficas arrastando e soltando os ícones de uma paleta de menus para a tela, permitindo também a edição destes ícones usando um editor de propriedades [B].

O software *Zero's Program* trata-se de um software de código aberto (open source). Os programas de código aberto possuem seu código-fonte aberto total ou parcialmente para a comunidade, permitindo melhorias e aprimoramentos [2]. Uma das plataformas que permite o uso deste feito é o *GitHub* [A], que permite que o desenvolvedor hospede seu projeto em um repositório, que pode ser compartilhado diretamente com outros usuários ou pode ser aberto para a comunidade. O desenvolvedor ainda pode ter controle das alterações feitas no código por meio de um sistema de controle de versão de projeto.

2 METODOLOGIA

Ao desenvolver o presente projeto, seguiu-se uma linha de raciocínio mostrada no fluxograma (Figura 1). O início ocorre com a motivação, dos presentes autores, no desenvolvimento de um *Software* livre para auxílio no ensino de métodos numéricos. Após, foi necessária uma pesquisa bibliográfica e refino de principais autores, de modo a, embasar e facilitar o desenvolvimento do programa (sua linha de raciocínio encontra-se na próxima seção). Com isso, criou-se um repositório no site *GitHub*, com o intuito de, compartilhar a lógica de programação e futuras versões que os autores ou a comunidade virem a produzir.¹

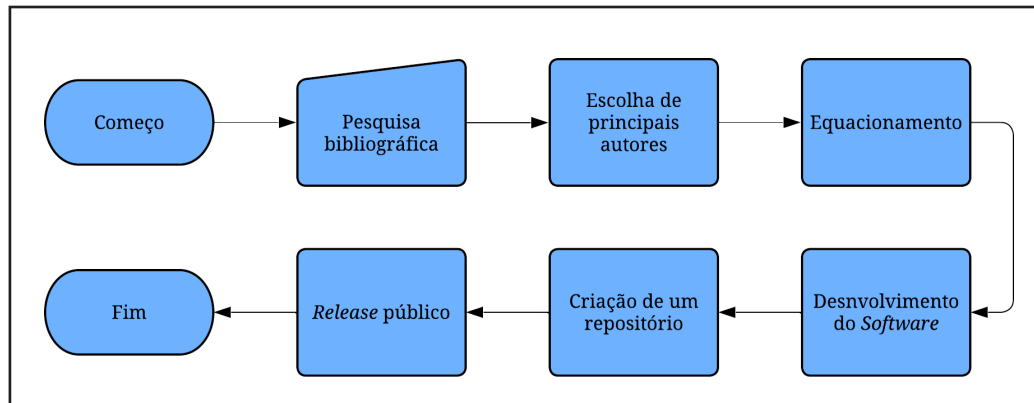


Figura 1. Fluxograma geral do trabalho.

3 DESENVOLVIMENTO DO SOFTWARE

Ao programar-se o Software, uma linha de raciocínio própria foi selecionada (Figura 2). No princípio, pesquisaram-se bibliotecas e recursos extras para o desenvolvimento da lógica de programação. Além disso, foi feito no papel um desenho simples de como a interface gráfica (GUI) do Software iria ficar. foi feito no papel. Detal maneira, teve-se início da programação, que foi dividida em duas partes em simultâneo, sendo elas, a criação da interface gráfica usando o Glade (local para inserir os dados de entrada) e a implementação das equações em Python (algoritmo de cálculo dos dados recebidos na entrada).

Com a implementação das equações, realizam-se vários testes comparativos com a literatura, para a averiguação das mesmas. Após, toma-se a decisão de continuar para a próxima etapa ou retornar para o equacionamento. Em simultaneidade, cria-se a GUI responsável pela distribuição de recursos de entrada, função analisada, intervalo de *plot* e análise, precisão e o número de iterações máximo (quantas vezes o método irá ser usado), e saída, valor encontrado para a raiz, precisão final e número de iterações usadas. Estando a construção da GUI em consonância com as equações do *back end*, o próximo passo consiste em averiguar-se o comportamento do *Software*, caso o mesmo não funcione de forma adequada, retorna-se para a etapa inicial de programação.

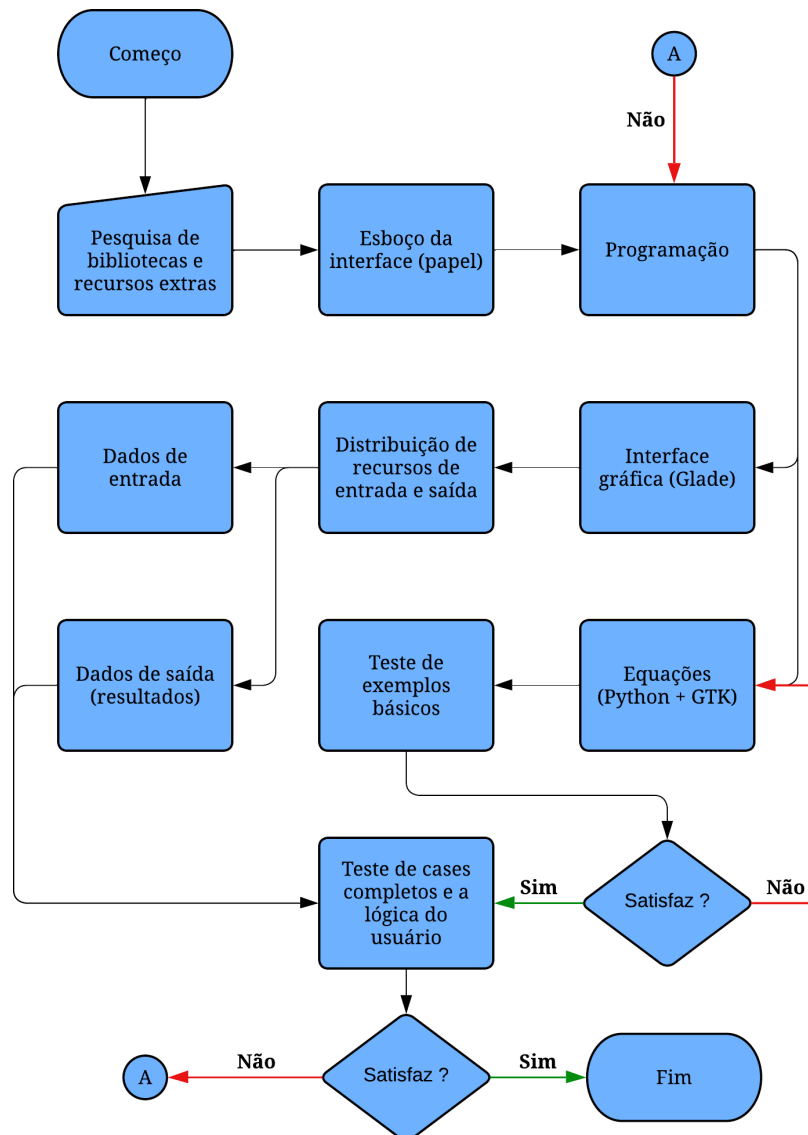


Figura 2. Fluxograma do desenvolvimento do Software.

4 RESULTADOS

Desse modo, foi obtida uma interface gráfica (Figura 3) simples e intuitiva, na tela inicial o usuário pode optar por selecionar os menus de navegação de início, ajuda e sobre. Na tela de início, foi selecionado para exemplificação de funcionamento, a equação $f(x) = x^3 - 9x + 3$. Logo após, foi feita uma análise gráfica de -5 a 5, para visualizar onde $f(x) = 0$ (função toca o eixo x com $y = 0$), posteriormente é escolhido o intervalo (0 a 1), precisão (0.0001), número de iterações (100) e métodos de cálculo, vale



destacar, que esse é um dos intervalos que contêm uma raiz, o que não representa o montante. Ao computar-se os dados, o programa calcula o valor encontrado para cada raiz, a precisão e o número de iterações usado para cada método (visualizar Figura 3 para melhor entendimento).

Métodos	Valor encontrado para a raiz	Precisão	Número de iterações usados
Bisseção	0.33758544921875	6.103515625e-05	14
Posição Falsa	0.337608955965838	2.22044604925031e-16	11
Newton Raphson	0.33760895596580065	5.628148057601479e-07	4
Birge Vieta	0.337608955965313	2.11835847518937e-6	3

Valor mínimo de x: -5 Valor máximo de x: 5

Função em termos de "x", f(x) =

☒ Bisseção ☒ Posição Falsa ☒ Newton Raphson ☒ Birge Vieta (somente para polinômios)

V(1.0)

Figura 3. Interface gráfica e exemplo de funcionamento do Software.

Inserindo-se dados de análise de forma incoerente, como o número de iterações menor que um, ponto anterior a raiz ser maior que o posterior a raiz (Figura 4), dentre outros, isso irá gerar uma mensagem de erro/aviso, notificando o usuário que algum erro específico fora cometido.

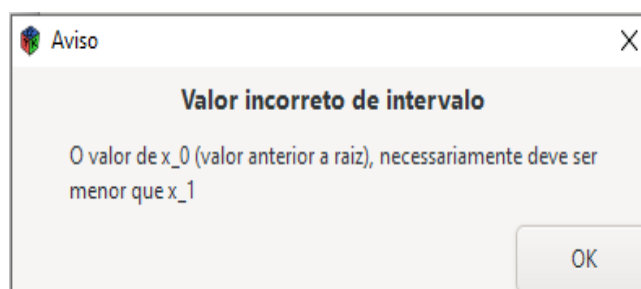


Figura 4. Mensagem de aviso.

Caso o usuário não conheça o comportamento da função selecionada, número de raízes e qual o intervalo que uma raiz se encontra, é possível gerar um gráfico (Figura 5) com um intervalo qualquer, clicando no botão “Plotar”.

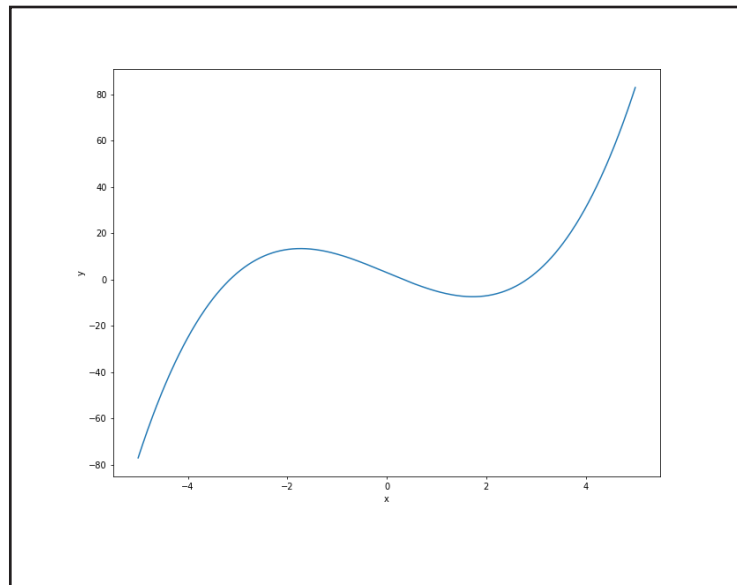


Figura 5. Gráfico da função selecionada.

Após criado o gráfico, o usuário deverá preencher os dados de entrada faltantes e clicar no botão “Calcular”. Poucos segundos depois, uma janela irá se abrir, mostrando os gráficos de convergência para cada método selecionado (Figura 6). Em paralelo, os valores encontrados para cada raiz, precisão final e número de iterações usadas, encontram-se no menu inicial.

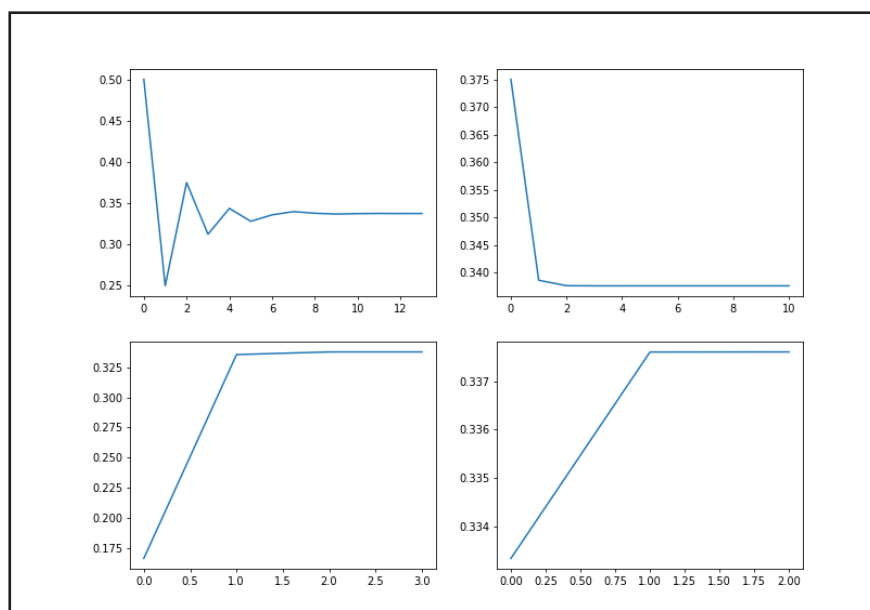


Figura 6. Gráfico de convergência para cada método.



5 CONCLUSÕES

Portanto, o desenvolvimento do programa *Zeros's Program*, se mostrou eficiente e acessível para a realização de cálculos complexos de zeros de funções. Através da disponibilidade gratuita e a colaboração da comunidade em geral, o *software* apresenta um grande potencial de modificações e melhorias, podendo gerar trabalhos futuros em diversas áreas, pois, o mesmo pode ser usado para ensino, análise e modificações, até mesmo, alterando a GUI do programa para ficar mais chamativa (bonita) e acessível para diferentes públicos. Se o presente leitor deseja testar o *Software* [D], ele se encontra disponível no GitHub e nas referências abaixo.

REFERÊNCIAS

- [1] CHAPRA, Steven C.; CANALE, Raymond P. (2016) *Métodos Numéricos para Engenharia - 7ª Edição*. McGrawHill Brasil.
- [2] GOONA, Nithin Kumar et al. (2023) DssPyLib: An open-source python FEM software to solve Poisson equation in 2-D using distributed source scheme. *SoftwareX*, v. 21, p. 101308,
- [3] JAIN, Mahinder Kumar. (2003) *Numerical methods for scientific and engineering computation*. New Age International.
- [1] PISTORI, H. Programação Multiplataforma Utilizando GTK.

REFERÊNCIAS WEB

- [A] GitHub, site. GitHub - Let's build from here. (2023) Disponível em: <https://github.com/> (último acesso em: 27/03/2023).
- [B] GLADE, site. Glade - A user Interface Designer. 2022. Disponível em: <https://glade.gnome.org/> (último acesso em: 20/03/2023).
- [C] GTK, site. GTK (GIMP Toolkit). (2022) Disponível em: <https://www.gtk.org/> (último acesso em: 20/03/2023).
- [D] Repositório do Zero's Program. (2023) Fonte: https://github.com/davidlp-eng/zeros_programa (último acesso em: 27/03/2023).

Sobre os autores:

David Lucas Pereira

Autor correspondente

Estudante de Engenharia Mecânica pela Universidade Federal de Santa Maria/UFSM

É membro do Grupo de Mecânica de Materiais e Estruturas (GMEC)

E-mail: david.pereira@acad.ufsm.br • <https://orcid.org/0000-0003-4309-4783>



Emerson Fouchy Santos

Estudante de Engenharia Mecânica pela Universidade Federal de Santa Maria/UFSM

É membro do Grupo de Mecânica de Materiais e Estruturas (GMEC)

E-mail: emerson.fouchy@acad.ufsm.br • <https://orcid.org/0000-0001-7816-7028>

Como citar este artigo

Pereira, D. L. & Santos, E. F. (2023) Desenvolvimento de um Software aberto para cálculo de zeros de funções. *JESTA*, Cachoeira do Sul, (2) e75178, Disponível em: <https://periodicos.ufsm.br/JESTA/article/view/75178>. Acessado em: dia mês abreviado. ano.